Metody inteligencji obliczeniowej a GGP

Karol Walędzik

Agenda

- *General Game Playing*?
- *General Game Players*
- MIO w GGP:
  - generowanie komponentów funkcji ewaluacyjnej
  - proces nauki

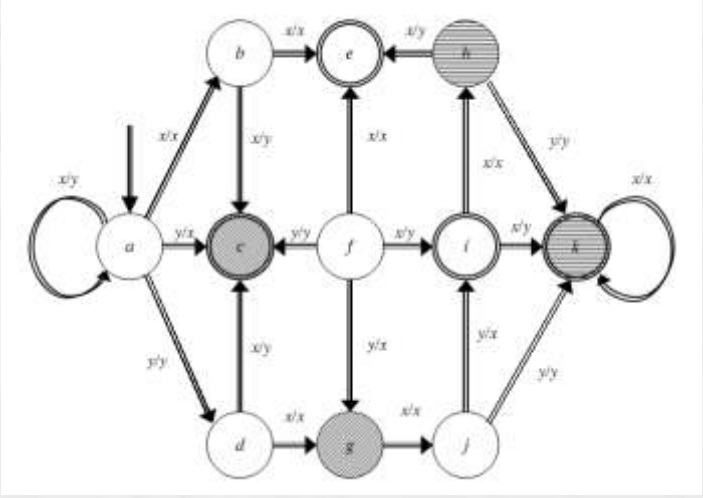
General Game Playing?

- Cel:
  - stworzenie systemu umiejącego grać/nauczyć się grać we „wszystkie” gry
- Turniej w ramach *AAAI National Conference*
  - corocznie od 2005 roku
- Przebieg rozgrywki w ramach turnieju:
  - prezentacja zasad i czas na ich analizę (np. 5m)
  - rozgrywka z ograniczeniem czasowym na wykonanie pojedynczego ruchu

Model gry

- Model gry:
  - skończona
  - skończona liczba graczy
  - synchroniczna
    - wszyscy gracze ruszają się równocześnie (ale dopuszczalne ruchy typu *noop*)
  - skończona liczba legalnych ruchów w każdym ze skończonej liczby stanów
  - zmiany stanu tylko w wyniku ruchów
  - ➔ maszyna stanowa

Model gry



M. Genesereth, N. Love, and B. Pell. General Game Playing: Overview of the AAAI Competition. AI Magazine, 26(2):62-72, 2005.

Game Definition Language

- Opis gry jako maszyny stanowej: zbyt rozwlekły
- Krótszy sposób opisu: Game Definition Language (GDL)
  - opis gry za pomocą fomuł logicznych
  - język bazujący na zmodyfikowanym Datalogu
    - a Datalog to podzbiór Prologa
  - podstawowe relacje:
    - *role, true, init, next, legal, does, goal, terminal*

GDL: Tic-Tac-Toe

```
 1. (role xplayer)
 2. (role oplayer)
 3. (init (cell 1 1 b))
 4. (init (cell 1 2 b))
 ...
 5. (init (cell 3 3 b))
 6. (init (control xplayer))
 7. (<= (next (cell ?m ?n x))
 8.     (does xplayer (mark ?m ?n))
 9.     (true (cell ?m ?n b)))
10. (<= (next (control xplayer))
11.     (true (control oplayer)))
 ...
13. (<= (row ?m ?x)
14.     (true (cell ?m 1 ?x))
15.     (true (cell ?m 2 ?x))
16.     (true (cell ?m 3 ?x)))
17. (<= (line ?x)
18.     (row ?m ?x))
 ...
19. (<= (legal ?w (mark ?x ?y))
20.     (true (cell ?x ?y b))
21.     (true (control ?w)))
22. (<= (legal xplayer noop)
23.     (true (control oplayer)))
 ...
24. (<= (goal xplayer 0)   (line o))
25. (<= (goal oplayer 100) (line o))
 ...
26. (<= terminal (line x))
```

J. Reisinger, E. Bahceci, I. Karpov, and R. Miikkulainen. Coevolving strategies for general game playing. In Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG 2007), 320-327, Honolulu, Hawaii, 2007. IEEE Press.

General Game Players

ClunePlayer

- Zwycięzca turnieju GGP w 2005
- Pomysł:
  - stworzenie heurystycznej funkcji ewaluacyjnej jako funkcji działającej na uproszczonym modelu gry
  - uproszczony model bazujący na 3 koncepcjach:
    - wypłata (*payoff*)
    - kontrola/mobilność (*control*)
    - terminalność/czas (*termination*)

FluxPlayer

- Zwycięzca turnieju GGP w 2006
- Pomysł:
  - stworzenie heurystycznej funkcji ewaluacyjnej na podstawie oceny stopnia spełnienia celów gry
    - wykorzystanie elementów logiki rozmytej

CADIAPlayer

- Zwycięzca turnieju GGP w 2007 i 2008
- Koncepcja:
  - rozwiązanie oparte na symulacjach Monte-Carlo
    - konkretniej: UCT
      - (Upper Confidence-bounds applied to Trees)
  - logika gry reprezentowana przez automatycznie generowany podprogram w Prologu

nnrg.hazel

- 6. miejsce (z 12) w turnieju GGP w 2006 roku
- Idea:
  - funkcja ewaluacyjna reprezentowana przez sieć neuronową
  - nauka z wykorzystaniem koewolucji
    - NEAT (NeuroEvolution of Augmenting Topologies)
      - jednoczesna ewolucja wag i architektury sieci neuronowej
      - rozpoczęcie procesu od bardzo prostej struktury sieci

Metody inteligencji obliczeniowej w GGP

MIO w GPP

- Podstawowe pytania (na początek):
  - algorytm wyboru ruchu:
    - algorytm *min-maxowy*
      - typowe usprawnienia:
        - tablice transpozycji
        - *history heuristic*
        - alg. alfa-beta obcięć

MIO w GPP

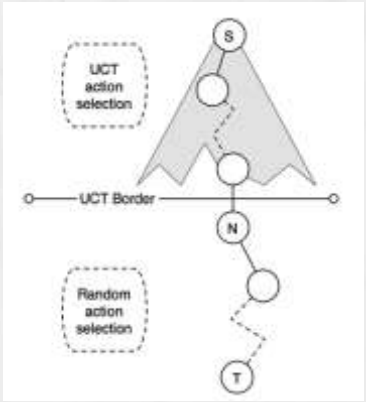
- Podstawowe pytania (na początek):
  - algorytm wyboru ruchu:
    - **UCT**

$$a^* = argmax_{a \in A(s)} \left\{ Q(s,a) + C\sqrt{\frac{\ln N(s)}{N(s,a)}} \right\}$$

      - Q(s,a) – średni dotychczasowy wynik pary stan-ruch
      - N – liczba wizyt w danym stanie/wykonań danej akcji
      - akcje nigdy nie wykonywane wybierane są w pierwszej kolejności

MIO w GPP

- Podstawowe pytania (na początek):
  - algorytm wyboru ruchu:
    - **UCT**



MIO w GPP

- Podstawowe pytania (na początek):
  - postać funkcji ewaluacyjnej (w przypadku UCT – opcjonalna):
    - **liniowa kombinacja parametrów**
      - jakich parametrów?
    - sieć neuronowa
      - wejście?

Komponenty funkcji ewaluacyjnej

Komponenty funkcji ewaluacyjnej

- Podejście podstawowe:
  - wyrażenia występujące w opisie gry
    - uogólniane przez podstawianie zmiennych za stałe i możliwych stałych za zmienne
    - cechy nie muszą być binarne
      - zliczanie sposobów rozwiązania zmiennych w wyrażeniu

```
cell(1, 1, x)        cell(1, 1, ?)
cell(?, ?, x)        cell(1, ?, x)
```

Komponenty funkcji ewaluacyjnej - Kuhlmann et al.

- Identyfikacja wzorców leksykalnych:
  - *następnik* (→ częściowy porządek):
    - `successor(a, b), successor(b, c) ...`
  - *licznik*:
    - `next(counter ?c) :- true(counter ?p), successor(?p, ?c)`
  - *plansza*
    - siatka 2D zmieniająca stan w czasie gry

Komponenty funkcji ewaluacyjnej - Kuhlmann et al.

- Identyfikacja wzorców leksykalnych:
  - *znacznik*
    - obiekty na polach planszy
  - *kamień*
    - znaczniki, które nigdy nie występują w więcej niż jednym egzemplarzu

Komponenty funkcji ewaluacyjnej - Kuhlmann et al.

- Komponenty na podstawie wzorców:
  - współrzędne kamieni
  - odległości między kamieniami
  - liczności, w tym liczba znaczników

Komponenty funkcji ewaluacyjnej - Cluneplayer

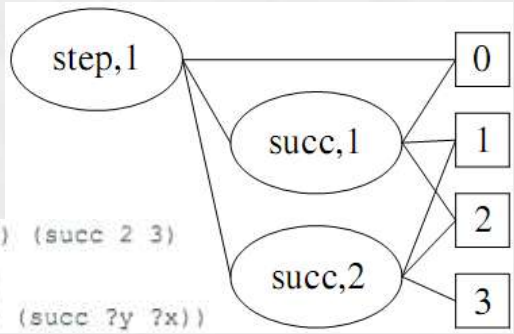
- Interpretacja podstawowych wyrażeń:
  - *liczność* – liczba sposobów rozwiązania zmiennych
  - *odległość* – bazująca na stwierdzeniu, że 2 symbole w relacji binarnej sąsiadują ze sobą
  - *częściowe rozwiązanie* – jaka część wyrażeń w koniunkcji jest spełniona

Komponenty funkcji ewaluacyjnej - Fluxplayer

- Semantyczna identyfikacja struktur (żeby zastosować niebinarne wartości atomów):
  - *następniki*
  - *uporządkowanie*
  - *liczności*
  - *plansza*
  - analiza częściowo poprzez generowanie hipotez i ich sprawdzanie

Komponenty funkcji ewaluacyjnej - Fluxplayer

- Wyznaczanie dziedzin argumentów predykatów/funkcji:
  - Graf zależności:




```
(succ 0 1) (succ 1 2) (succ 2 3)
(init (step 0))
(<= (next (step ?x))
    (true (step ?y)) (succ ?y ?x))
```

Komponenty funkcji ewaluacyjnej – nnrg.hazel

- Losowa projekcja wyrażeń na 40 neuronów wejściowych
  - niekoniecznie wzór do naśladowania …

Komponenty funkcji ewaluacyjnej – plany

- Na początek plan minimum:
  - wygenerowanie możliwych wyrażeń
    - w tym podstawianie zmiennych/stałych
      - identyfikacja dziedzin jak we Fluxplayer
      - → liczności
        - w tym liczności mieszane: wystąpienie elementu w iluś pozycjach o tej samej dziedzinie
  - wstępne filtrowanie: stabilność, korelacja z wynikiem końcowym
    - mierzona za pomocą symulacji

Komponenty funkcji ewaluacyjnej – plany

- Pierwsze uzupełnienie:
  - koncepcja następnika, porządku i odległości (między faktami, a nie atomami):
    - `następnik(A, B)` ⇔ istnieje implikacja, w której **A** jest jedną z przesłanek, a **B** następnikiem

    lub jeszcze lepiej:
    - `następnik(A, B)` ⇔ istnieje stan, w którym zachodzi **A** i istnieje ruch, który doprowadzi do stanu, w którym **B**
- Wstępne założenie:
  - brak szukania analogii z typowymi grami (np. *plansz*)

Nauka

- Możliwości:
  - koewolucja
  - TD(λ)
  - **podejście warstwowe**
    - w komplecie z dowolnym podejściem *Supervised Learning*

Podejście warstwowe do nauki

- **Faza I –** generowanie funkcji ewaluacyjnej dla ostatniej fazy gry:
  - wygeneruj pozycje bliskie końca gry
  - oceń je za pomocą algorytmu *min-maxowego*
  - przeprowadź naukę z nadzorem
- **Faza *n***:
  - uzupełnij zbiór uczący o nieco wcześniejsze pozycje
  - oceń je wykorzystując funkcję ewaluacyjną uzyskaną w fazie *n-1*
  - przeprowadź naukę z nadzorem

„Przeprowadź naukę z nadzorem” - plany

- Na początek alg. ewolucyjny
  - osobnik = wybrane komponenty + ich wagi
  - mutacja
    - jeden z operatorów podmienia/usuwa/dodaje komponenty
      - wpływ na decyzję o usunięciu: prosta analiza wrażliwości
      - wpływ na decyzję o dodaniu: stabilność, korelacja z wynikiem
  - funkcja przystosowania:
    - odwrotność błędu na zbiorze uczącym
    - specjacja
    - delikatna preferencja dla prostszych osobników (mniej komponentów)

Najważniejsze źródła

- James Clune: Heuristic Evaluation Functions for General Game Playing. AAAI 2007: 1134-1139
- Hilmar Finnsson, Yngvi Björnsson: Simulation-Based Approach to General Game Playing. AAAI 2008: 259-264
- M. Genesereth, N. Love, and B. Pell. General Game Playing: Overview of the AAAI Competition. AI Magazine, 26(2):62-72, 2005
- G. Kuhlmann, K. Dresner, and P. Stone: Automatic heuristic construction in a complete General Game Player. In Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence (AAAI-06), pages 1457{1462, Boston, MA, 2006. AAAI Press.
- M. Kusiak, K.Walędzik, and J. Mańdziuk. Evolution of heuristics for give-away checkers. In W. Duch et al., editors, Artificial Neural Networks: Formal Models and Their Applications - Proc. ICANN 2005, Part 2, Warszawa, Poland, volume 3697 of LNCS, pages 981-987. Springer, 2005
- J. Reisinger, E. Bahceci, I. Karpov, and R. Miikkulainen. Coevolving strategies for general game playing. In Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG 2007), 320-327, Honolulu, Hawaii, 2007. IEEE Press
- Stephan Schiffel, Michael Thielscher: Fluxplayer: A Successful General Game Player. AAAI 2007: 1191-1196